

Solving the minimum vertex floodlight problem with hybrid metaheuristics

Antonio L. Bajuelos^{1,4}, Santiago Canales^{2,5}, Gregorio Hernández^{3,5},
Mafalda Martins^{1,6}

¹ Universidade de Aveiro, Portugal
leslie@ua.pt, mafalda.martins@ua.pt

² Universidad Pontificia Comillas de Madrid, Spain
scanales@dmc.icae.upcomillas.es

³ Universidad Politécnica de Madrid, Spain
gregorio@fi.upm.es

Abstract. In this paper we propose four approximation algorithms (metaheuristic based), for the Minimum Vertex Floodlight Set problem. Urrutia et al. [9] solved the combinatorial problem, although it is strongly believed that the algorithmic problem is \mathcal{NP} -hard. We conclude that, on average, the minimum number of vertex floodlights needed to illuminate a orthogonal polygon with n vertices is $\lceil \frac{n}{4.29} \rceil$.

Introduction

In this paper we address the Minimum Vertex $\frac{\pi}{2}$ -Floodlight Set problem (MVF(P) problem). This problem asks for the minimum number of vertex $\frac{\pi}{2}$ -floodlights necessary to illuminate a given orthogonal simple polygon P with n vertices (n -ogon, for short [8]). A vertex $\frac{\pi}{2}$ -floodlight is a source light with an angle of illumination of $\frac{\pi}{2}$ placed on a vertex of an n -ogon. Since this paper only deals with $\frac{\pi}{2}$ -floodlights, and for simplicity, the term *floodlight* is used instead of “ $\frac{\pi}{2}$ -floodlight”. It is assumed that the vertex floodlights are edge-aligned and that each reflex vertex has at most two vertex floodlights. Urrutia [9] proved that $\lfloor \frac{3n-4}{8} \rfloor$ vertex floodlights are occasionally necessary and always sufficient to illuminate a n -ogon. But for many n -ogons this number is clearly too large. This fact justifies the algorithmic MVF(P) problem. It is strongly believed that this problem is \mathcal{NP} -hard. A way to deal with this computational complexity is to develop approximation algorithms to tackle the problem. In general, these approximation methods can be designed specifically to solve the problem (e.g., *greedy strategies*) or can be based on general metaheuristics (e.g., *Simulated Annealing* (SA) and *Genetic Algorithms* (GAs)). There are several works where non-metaheuristic based approximation algorithms were developed to tackle art gallery problems (e.g., [2, 5, 6, 8]). Recently, some work has been made on the application of metaheuristic techniques for these problems (see [1, 3]).

Our contribution: We present four approximation algorithms, based on general metaheuristics, to tackle the MVF(P) problem. Since the optimal solution to the MVF(P) problem is unknown, we developed a method that allows us to determine a lower bound for our algorithms, as in [2]. In this way, we are able to find the approximation ratio of our strategies. Our experiments were performed on a large set of randomly generated orthogonal simple polygons.

⁴Supported by FCT through CIDMA of University of Aveiro.

⁵Partially supported by Projects MTM2008-05043 and HP2008-0060.

⁶Supported by FCT through CIDMA of University of Aveiro and grant SFRH/BPD/66431/2009.

1 Approximation methods

A set of vertex floodlights for an n -ogon P is a *vertex floodlighting set* for P if they illuminate P . We denote a vertex floodlighting set for P by F and its cardinality by $|F|$. Since the existence of an efficient algorithm to determine a minimum-cardinality vertex floodlighting set is unknown, we developed four approximation algorithms to tackle this problem. The first is based on the SA metaheuristic, called M_1 ; the second is based on the GAs metaheuristic, named M_2 , and the last two are hybrid algorithms, designated by M_3 and M_4 .

Simulated Annealing Strategy (M_1). A *configuration* is a chain with length $n + r$ (the reflex vertices are duplicated to determine the two possible positions of the floodlight), where the value of each element is 0 (floodlight-“on”) or 1 (floodlight-“off”); see Figure 1.

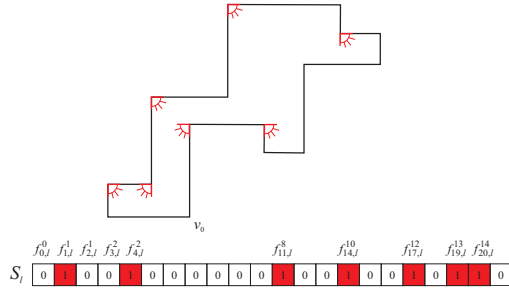


FIGURE 1. A configuration for a 16-ogon.

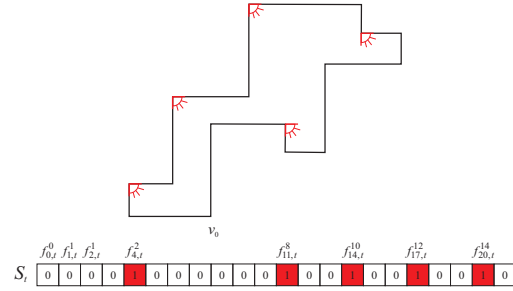


FIGURE 2. Initial configuration.

The *objective function* assigns to each configuration its number of 1's. A *neighbour* of a configuration is generated by switching from 0 to 1, or vice versa, a randomly chosen element. To generate the *initial configuration*, we used the top-left rule [9], that guarantees the illumination of P (see Figure 2). We performed a comparative study taking into account three different *initial temperatures* T_0 : (1) $T_0 = n$; (2) $T_0 = n/4$ and (3) $T_0 = 500$. Concerning the *temperature decrement rule*, we made an analysis on three different types of rules: (1) $T_{k+1} = T_0/(1 + k)$; (2) $T_{k+1} = T_0/e^k$ and $T_{k+1} = 0.9 \times T_k$. The *number of iterations in each temperature* T_k is equal to $\lceil T_k \rceil$. Finally, the *termination condition* consists in finishing the search when $T_k \leq 0.005$ or when during 3000 consecutive series of temperatures no new best solution is obtained and the percentage of accepted solutions is less than 2%.

Genetic Algorithms Strategy (M_2). An *individual* is represented by a chain with length $n + r$, where the value of each gene is 0 or 1. We choose the *population size* to be $\lfloor \frac{3n-4}{8} \rfloor$. To create the *initial population*, we generate each of the $\lfloor \frac{3n-4}{8} \rfloor$ individuals in the following way: all of its genes are set to 1; then a gene is randomly selected and its value is set to 0 if the resultant individual is valid; otherwise its value remains 1. The *fitness function* is defined as the number of 1's in each individual. We used the tournament selection method to the genetic operator *selection* and a variant of the single point crossover to the *crossover*. The *mutation* step is relatively simple: for each binary digit it merely flips it from 0 to 1 or vice versa (if the obtained individual is not valid, it is rejected). To *generate a new population*, the worst individual is replaced by the child

obtained at mutation. In order to *evaluate a population*, we consider its fitness as the minimum value of the fitness function when applied to its individuals. Finally, we *stop the search* when the fitness of the population remains unchanged for 500 generations.

Hybrid Strategies (M_3 and M_4). GAs and SA are population-based and single-solution search methods, respectively. Different combinations of these two types of metaheuristics have provided powerful search algorithms. These combinations are known as hybrid metaheuristics [7]. To solve the MVF(P) problem, we developed two different hybrid metaheuristics, that fundamentally use a genetic algorithm. However, in the first method, M_3 , for the initial population of the genetic algorithm we generate $\lfloor \frac{3n-4}{8} \rfloor$ individuals, running $\lfloor \frac{3n-4}{8} \rfloor$ times the SA metaheuristic. In the second method, M_4 , in addition to the classical crossover and mutation operators, we add a new genetic operator based on the SA metaheuristic. Basically the process consists of applying the SA after the crossover operator, in order to refine the solution produced by that operator. After this operation, the mutation operator is applied. Since SA is a genetic operator, it occurs with probability p_{sa} .

Since the optimal solution for the MVFL(P) problem is unknown, we developed a method to compute a lower bound on the optimal number of vertex floodlights for each instance in the performed experiments. For that, we used the notion of *floodlight visibility-independent set*, which is a finite set of points on an n -gon P , $FIS \subset P$, such that, for all $p, q \in FIS$, p and q are not illuminated by the same floodlight. It can be concluded that the number of points on a maximum-cardinality floodlight visibility-independent set is a lower bound for the optimal number of vertex floodlights on P . However, as far as it is known, the existence of an efficient algorithm to determine this lower bound is unknown. Therefore, we developed a greedy algorithm to find large floodlight visibility-independent sets, which we designated by A_1 .

2 Experiments and results

The implementation of our algorithms was done in C/C++ (for MS Visual Studio 2005) on top of CGAL 3.2.1 [4]. The above described methods were tested on a PC featuring an Intel(R) Core (TM)2 CPU 6400 at 2.66 GHz and 1 GB of RAM. We performed extensive experiments with the strategies described in the previous section on a large set of randomly generated orthogonal polygons. To generate these polygons, we used the polygon generator developed by O'Rourke (personal communication, 2002). In this section we present our results and conclusions from the experiments. According to Section 1, there are several choices for two of the SA parameters: T_0 and the temperature decrement rule. The different combinations of their values give rise to nine cases. We analyzed these nine cases by comparing the number of vertex floodlights $|F|$, the runtime (in seconds) and the number of iterations performed by each of them. We carried out a statistical study to compare the results obtained by them, but due to lack of space we omit its details. In this study, we concluded that: (i) concerning $|F|$, the case where $T_0 = 500$ and $T_{k+1} = T_0/(1+k)$ is the best one. Therefore, this was the case considered as the SA strategy, i.e., method M_1 ; (ii) regarding runtime, the case where $T_0 = n/4$ and $T_{k+1} = T_0/e^k$ is the fastest algorithm and, although the returned number of vertex floodlights is worse, it can be used in the hybrid methods. So we used it to generate the initial population in M_3 and as genetic operator in M_4 . Then we analyzed and evaluated the

results obtained with our four methods. Table 1 presents the obtained results (averages of 40 n -ogons each one).

n	M_1			M_2			M_3			M_4		
	$ F $	Time	Iterations	$ F $	Time	Iterations	$ F $	Time	Iterations	$ F $	Time	Iterations
30	7.75	13.85	4916.80	8.47	5.70	297.875	7.80	7.15	520.15	6.97	11.52	699.50
50	12.52	39.85	6436.30	14.20	37.37	1071.10	12.27	28.05	580.60	11.62	44.17	683.85
70	17.90	89.22	8295.60	19.97	105.75	1160.30	17.52	78.70	661.90	16.25	127.80	893.60
100	25.12	185.10	10202.00	29.05	290.35	2278.30	25.57	243.20	754.55	23.30	373.55	1061.50
110	27.50	230.62	11049.00	31.40	357.87	1954.80	27.97	324.65	677.77	25.00	485.30	1052.70
130	33.00	340.52	12683.00	37.30	531.40	2360.10	33.30	547.20	783.57	30.12	747.92	1158.50
150	37.65	441.57	13661.00	42.02	763.60	3284.60	38.47	842.50	843.05	34.70	1078.10	1241.40
200	50.25	871.90	17972.00	56.82	1584.50	4568.10	52.22	2163.30	919.05	46.70	2660.20	1520.60

TABLE 1. Results obtained with M_1 , M_2 , M_3 and M_4 .

We performed a statistical analysis to check the differences between the solutions obtained with them (again, we omit its details), and we concluded that: concerning the obtained solutions, the hybrid method M_4 is the best one and the method M_2 the worst one. The methods M_1 and M_3 can be considered equal. Consequently, we continued our study considering M_4 as the best strategy. To infer about the average of the minimum number of vertex floodlights needed to illuminate an orthogonal polygon, we used the least squares method and the following linear adjustment was obtained, with a correlation factor of 0.99: $f(x) = 0.2328x - 0.1091 \approx \frac{x}{4.29}$. Thus, it can be concluded that, on average and approximately, the minimum number of vertex floodlights needed to illuminate an n -ogon was observed to be $\lceil \frac{n}{4.29} \rceil$. In order to get a quantitative measure on the quality of the calculated $|F|$, the floodlights visibility-independent sets were computed on our instances (the eight sets of polygons described above). The ratio between the smallest F (obtained with M_4) and the largest FIS (obtained with A_1) never exceeded 2, which implies that algorithm M_4 has an approximation ratio less than or equal to 2.

References

- [1] M. Abellanas, E. Alba, S. Canales and G. Hernández, Solving the illumination problem with heuristics, in: *Numerical Methods and Applications*, Vol. 4310 of LNCS, Springer (2006), 205–213.
- [2] Y. Amit, J. S. B. Mitchell and E. Packer, Locating Guards for Visibility Coverage of Polygons, in: *Proceedings of the Workshop on Algorithm Engineering and Experiments*, 2007, 1–15.
- [3] A. L. Bajuelos, A. M. Martins, S. Canales and G. Hernández, Metaheuristic approaches for the minimum vertex guard problem, in: *The Third IEEE International Conference on Advanced Engineering Computing and Applications in Sciences*, IEEE Computer Society (2009), 77–82.
- [4] CGAL, Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [5] S. Ghosh, Approximation algorithms for art gallery problems in polygons, in: *Discrete Applied Mathematics*, **158** (2010), 718–722.
- [6] M. C. Couto, C. C. Souza and P. J. Rezende, An exact and efficient algorithm for the orthogonal art gallery problem, in *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing*, Washington, DC, USA, IEEE Computer Society, 2007, 87–94.
- [7] E. G. Talbi, A taxonomy of hybrid metaheuristics, *Journal of Heuristics* **8** (2002), 541–564.
- [8] A. P. Tomás, A. L. Bajuelos and F. Marques: Approximation algorithms to minimum vertex cover problems on polygons and terrains, Vol. 2657 of LNCS, Springer (2003), 869–878.
- [9] J. Urrutia, Art gallery and illumination problems, in: J.-R. Sack and J. Urrutia, eds., *Handbook of Computational Geometry*, Elsevier, 2000, 973–1027.